

METHODS AND SYSTEMS FOR SHARING DATA

Reference to Related Application

[0001] This application claims the benefit under 35 U.S.C. §119 of U.S. patent application Nos. 60/472,966 entitled "SYSTEM AND
5 PROCESS FOR SENDING ELECTRONIC MESSAGING ATTACHMENTS" filed 22 May 2003 and 60/319,701 entitled "SYSTEM AND PROCESS FOR SENDING ELECTRONIC MESSAGING ATTACHMENTS", filed 15 November 2002, both of which are hereby incorporated by reference herein.

10

Technical Field

[0002] This invention relates to sharing electronic data in computerized environments. Embodiments of the invention provide methods and systems for sharing selections of electronic files and/or
15 folders. Methods and systems according to preferred embodiments of the invention are suitable for use in contexts where security, privacy and convenience are all important.

Background

20 [0003] In a computerized environment, many activities require sharing of data. Shared data may include: documents, images, video, audio, database records as examples. Data is often organized using electronic files or records in databases. These data containers are typically kept in an electronic storage facility accessible by a computer
25 system. Owners of data typically require that their data be kept secure and private to themselves. If the data is to be made available to others then access is limited to those others for whom access has been pre-authorized.

30 [0004] There are many examples in the prior art of computerized systems for maintaining security and privacy of data. Some examples include:

- systems which restrict physical access to storage, computer and network systems;
- systems which restrict communication access to computer and network systems to those with password-protected user accounts;
- 5 and,
- systems which restrict access to electronic files or data processing applications using access control lists, passwords, or other authenticating mechanisms.

10 [0005] Once the intent to share data with others exists, the key steps to effect sharing are:

- a sharer makes the data available for sharing;
- the sharer communicates the intent to share data with recipients;
- and,
- 15 • the recipients obtain the shared data.

Sharing data presents challenges of security, privacy and convenience. These challenges vary depending on the computerized environments of the sharer and the recipients.

20 [0006] Making data available for sharing can involve many challenges. A major challenge is the effort and complexity required to selectively circumvent the security and privacy mechanisms provided by the sharer's computerized environment so that a recipient can access the shared data. Depending on the degree of security or privacy a sharer

25 desires for the data, the effort and complexity can be minor or significant. At one extreme, the sharer can publish the data on a publicly available storage facility, such as an internet web server. This entails obtaining the privilege to post information, copying the information to a location known to the web server and configuring the

30 application to make the data available. Additional security can be provided by protecting the data with a password, but additional effort

must be expended to secure the data with a password. Passwords are often easily lost, misinterpreted by the recipient or created in a manner that is easily guessed. At another extreme, the sharer can iteratively add the intended recipients to a list of those entitled to access the original data. The effort for this extreme can be protracted as additional recipients are identified later. The complexity increases as recipients identify others that need to share the data, as the sharer may need to be contacted to authorize access.

10 [0007] Another challenge is that intended recipients may have no physical or communication access with the computer system storing the shared data. Again, tradeoffs between convenience and security may be made. In some cases the sharer does not have the authority to make those tradeoffs. For example, when private data is stored on a sharer's computer inside a company's local network, the company's network administrator likely will not permit an intended recipient, using a computer outside the local network, to have communication access with the local network or the sharer's computer. Even if access is permitted, the methods for enabling this are typically complex, involving establishing user accounts, configuring physical access and communication access.

25 [0008] Another challenge is that the shared data may not be static or may not be statically located. For example, the sharer may intend to share a specific version of a document while continuing to edit it. This challenge may be addressed by making a copy of the shared data. The copy must then be stored in a place where it can be later located by a recipient. This can be problematic as a sharer must consume additional storage resources, may inadvertently move or remove the copy, or may forget to remove the copy after it has been shared so that resources are not consumed longer than necessary.

[0009] Another challenge is that the shared data may actually be a selection of independent data items. Editing access control lists for a large collection of files can be a time consuming and error prone task.

5

[0010] Assuming data is available for sharing, challenges can arise with communicating the intent to share. Communication can take many forms, examples including: physical delivery of information, audio or video communication, and electronic communication. In a computerized environment, physical delivery is sometimes used but the volume of information that can be transmitted in this manner is limited to the storage capacity of the media that is physically delivered. Audio and video communication is suitable for some forms of information but is also limited in volume by the bandwidth of the communication channel.

10

Electronic communication is suitable for most forms of information that can be digitized but also suffers from communication channel bandwidth limitations. Electronic communication is perhaps the most common form in a computerized environment, with email and instant messaging being amongst the most popular modes of electronic communication. If the communication only includes information about the intent to share, all of these forms of communication are suitable.

15

20

[0011] It is common in the prior art for shared data to be included in communications which convey the intent to share. This is often done to overcome the challenges with making data available for sharing. In short, the sharing setup activities are simplified by adding the data to an existing communication. The tradeoff made with this approach is that including shared data in, for example, e-mail messages, consumes resources, often at an inopportune time, proportional to the number of intended recipients. Communications resources are most severely affected and unfortunately are affected at a time of the sharer's

25

30

choosing. For example, a large file attached to an email, delivered to twenty people, may require that communication channels between 20 or more computer systems must immediately allocate capacity for the large message. Data storage resources are also affected when the data is
5 included in the communication, which can be problematic if only some of the recipients need the information or if some recipients need only part of the information. The storage problem extends beyond the sharer and recipient to other systems, such as email servers, that store copies of the data for each recipient in temporary and permanent user
10 communication archives.

[0012] The prior art includes a number of approaches for addressing some of these challenges. One approach, illustrated by data compression, enables a selection of data files to be compressed into a
15 single file (e.g. a zip file). The compressed data file may then be delivered through a file serving application or delivered in a communication. In both cases, the entire selection of data is communicated to the recipient. In the first scenario, a tradeoff between security, privacy and convenience is made by the sharer when
20 determining the method for making the data available. In addition, the location of the shared data is fixed at a location, making reorganization of the shared data storage facility difficult without affecting the convenience of the sharer and/or the recipients. In the second scenario, the sharer establishes his convenience as a priority relative to recipient
25 convenience and resource consumption.

[0013] Another approach, illustrated by Lamming et al. (US patent 5,862,321), teaches a tightly integrated data sharing system aimed at optimizing portable device resources and document security. The
30 system of Lamming et al. includes:

- a database of documents and tokens, each token providing a compact reference to a document in the database;
 - a set of devices, assigned to users that are configured in the system;
 - 5 • a document handling subsystem for users to create new documents and their tokens, and exchange tokens for documents at a time of a recipient's choosing; and
 - a security system for authenticating users as trusted users of the system.
- 10 This approach addresses some of the challenges described above but has limitations. A major limitation is that a recipient must be authenticated by the system. The time, expense and effort to establish broad-based user authentication in such a system will be a challenge, especially when the need for sharing with a user is unexpected or infrequent. Another
- 15 limitation is that a separate token is required for each document. Another limitation is that documents are referenced by including a storage address (e.g. a URL) in the token. This means that a document, referenced by a token, cannot be relocated without the token being regenerated.
- 20
- [0014]** Lambert et al. (PCT patent application WO 00/75779 A2), teach a tightly integrated data processing system aimed at using data tokens to reference and control user manipulation of data. Lambert et al. provide:
- 25 • a method for storing data objects to be processed;
 - a method for generating a token containing information about the data object, and what operations on the data object are permitted; and,
 - a method for a recipient to operate on a data object referenced by
- 30 a token.

This approach addresses some of the challenges discussed but relies on authentication schemes for security and privacy, which are less convenient for end users. Lambert et al. also describe a method of associating a token with one data object. Lambert et al. discuss
5 redactability of data objects but this relates to the sharer deciding which portions of the data object are available to specific recipients.

[0015] Another approach, illustrated by peer-to-peer file sharing systems, teaches a loosely integrated file sharing system. These systems
10 trade off security for convenience. Shared data is publicly available and easily discovered.

[0016] As discussed there are a variety of methods and systems in the prior art that attempt to address the various challenges associated
15 with sharing data while maintaining the security of the data and convenience for the users. Each of these approaches makes a trade off at the expense of one characteristic or another. Thus, a need exists for methods and systems to enable sharing that provide a better balance of security, privacy and convenience.

20

Summary of the Invention

[0017] This invention provides methods and systems for sharing data. Some preferred embodiments provide secure and private means of sharing data without compromising convenience and resource
25 utilization. A system according to one embodiment of the invention includes:

- a mechanism which a sharer can use to select data (e.g. files and folders) to be shared. The mechanism may be provided by an application in an operating system or a system component called a
30 tokenizer herein.

- a bundle server that stores a selection of data in a storage container, which may be called a bundle. The bundle server assigns the bundle an identifier that is substantially unguessable, and provides a mechanism for retrieving a bundle when presented with the bundle identifier corresponding to the bundle.
- a tokenizer that produces a token that represents the bundle. The token includes, among other things, the bundle identifier. The token can be delivered to a recipient by any suitable method (e.g. e-mail attachment).
- a redeemer that interacts with a bundle server to retrieve some or all of the contents of the bundle corresponding to the bundle identifier in a token. The redeemer makes the shared data available, for example by creating copies of the data in a storage facility or providing the data to an application. A recipient can use the redeemer to redeem a token at a time of his/her/its choosing.

[0018] In some embodiments of the invention a sharer can create a token corresponding to a selection of related data. As an example, the selection may include a set of documents and files relating to an activity. The documents and files in the selection may be of different types. Contextual information about the selection of data or an element of the selection of data may additionally be included in the token for the recipient's benefit. As an example, a sharer may wish to provide annotations about the selection of files or about specific files.

[0019] Some embodiments of the invention permit a recipient to selectively retrieve only the portions of the data from the container that are relevant to the recipient. This allows the recipient even greater convenience in determining how and when resources will be consumed.

- [0020]** In some embodiments of the invention, a storage container is freely available to anyone who possesses its token. Thus, there is no incremental effort required to share information with a new recipient. However, the storage container may only be accessed by presenting its identifier, whose value is completely unrelated to the content it holds, the sharer that generated it or the location where it is stored. The range and distribution of possible identifiers is structured so that it is prohibitive to try and guess a valid identifier for a storage container.
- [0021]** The storage container may be portable. In some embodiments of the invention, if the container has been moved or the computer system providing access to the storage container has changed then searching methods are used to locate the storage container in alternative storage locations.
- [0022]** Systems according to the invention may be constructed so that they can share data without modification to normal network security policies. Such systems may include a public communication relay service that facilitates communication between storage location and recipient computer systems, located on separate computer sub-networks, each of which do not permit unauthorized communications originating from outside their sub-network.
- [0023]** Systems according to the invention may have a loosely coupled architecture. This reduces the setup time required to implement such systems on a small scale (e.g. a few users sharing a few files in a local network) while enabling larger scale use (many users sharing many files across many sub-networks) to also be practiced with the same system without requiring incremental setup effort by the end users.

[0024] These and other aspects of the invention and features of embodiments of the invention are illustrated in greater detail in the detailed description.

5 Brief Description of Drawings

[0025] In drawings which illustrate non-limiting embodiments of the invention:

Figure 1 is a schematic diagram representation of a plurality of interconnected computer systems according to one embodiment of the
10 invention.

Figure 2 is a flowchart illustrating a method for sharing data according to one embodiment of the invention.

Figure 3A is a block diagram illustrating one system according to the invention which includes two computer systems.

15 Figure 3B is a block diagram illustrating another system according to the invention which includes several computer systems.

Figure 4 is a data structure diagram for a bundle according to one embodiment of the invention.

20 Figure 5 is a data structure diagram for a token according to one embodiment of the invention.

Figure 6 is a block diagram illustrating a system including a relay service according to one embodiment of the invention.

Detailed Description

25 [0026] Throughout the following description, specific details are set forth in order to provide a more thorough understanding of the invention. However, the invention may be practiced without these particulars. In other instances, well known elements have not been shown or described in detail to avoid unnecessarily obscuring the
30 invention. Accordingly, the specification and drawings are to be regarded in an illustrative, rather than a restrictive sense.

[0027] The invention provides methods and systems for sharing data between entities. Systems, according to a preferred embodiment, provide methods for a sharer entity to make data available for sharing
5 by recipient entities. Such systems generate a token representing data to be shared. The token contains information which can be used to identify a storage container holding information about the data to be shared. A sharer can deliver the token to intended recipients by any suitable method.

10

[0028] Systems, according to some embodiments, allow any entity possessing a token to retrieve the contents of the storage container corresponding to the token. The data to be shared may comprise electronic files stored by a computer system. In other embodiments,
15 data to be shared may include data organized by schemes other than a file system scheme. Examples include: data streams, data records, and distributed data records. The entities sharing information are most typically people interacting with computer systems. In other embodiments, intelligent entities, other than people, may perform the
20 roles of sharer and/or recipient. Examples of other entities that may play the role of sharer or recipient include software applications and programmable logic devices.

[0029] Figure 1 is a schematic diagram illustrating a plurality of
25 interconnected computer systems **100**, corresponding to one embodiment of the invention. Blocks **102A**, **102B** and **102C** (collectively blocks **102**) represent computer systems. Blocks **104A**, **104B**, and **104C** (collectively blocks **104**) represent parts of a computer network that provides communication connections between computer
30 systems **102A**, **102B** and **102C**. Blocks **104** may comprise sub-networks.

[0030] One embodiment of the invention uses two computer systems **102A** and **102B**. Block **102A** is a computer system used by a person that intends to share files with others. Block **102B** is a computer system used by an intended recipient of the files to be shared. Block
5 **102C** represents one or more other computer systems which may be connected to network **104** and may be included in systems according to other embodiments of the invention, as described below.

[0031] Each computer system **102** has a processing unit **112** and a
10 user interface comprising one or more output devices and one or more input devices. In the illustrated embodiment output devices comprise graphic display monitors **114** and input devices comprise mice **116** and keyboards **118**. Each processing unit **112** has access to a data store **110**, which is accessible to computer system **102** and may be part of the
15 computer system **102**.

Basic Aspect

[0032] Figure 2 is a flow chart illustrating a method **200** for sharing files according to a simple embodiment of the invention. Figure
20 **3A** shows a system **300A** according to one embodiment of the invention in which method **200** may be practised. Method **200** involves interactions of two computer systems **102A** and **102B**.

[0033] Method **200** begins when a sharer intends to make files
25 available for sharing. In block **204** the sharer interacts with tokenizer **316** running on computer system **102A** to identify a file selection **310** to be shared from a data store **110A** (or any other data store **110** accessible to computer system **102A**). Tokenizer **316** provides a user interface to present a display of folders and files from data store **110A**. The sharer
30 selects files and folders from data store **110A** to identify file selection **310**. File selection **310** can include any reasonable number of files

and/or folders to be shared. The files may be of diverse types. By way of example, only the files may include word processing documents, spreadsheets, graphics files, video or audio files, markup language files such as HTML or XML files, executable files, and/or text files.

5

[0034] In block 206, tokenizer 316 requests bundle server 324 to store information corresponding to file selection 310. Stored information may include meta-data about the selected files and folders as well as the contents of the selected files. In this illustration bundle server 324 is running on computer system 102A.

[0035] Bundle server 324 organizes information about shared file selections into at least one bundle store 320 located in data store 110A. Bundle store 320 holds at least one bundle. Each bundle stores information about a corresponding file selection 310. Each bundle can correspond to a specific request to make a file selection available for sharing.

[0036] In response to the request, bundle server 324 creates new bundle 322 in a bundle store 320. Bundle server 324 stores the information provided by tokenizer 316 in bundle 322. Bundle server 324 supplies information about bundle 322, bundle store 320, and bundle server 324 to tokenizer 316.

[0037] In block 208, in response to the receipt of information from bundle server 324, tokenizer 316 creates a new token 314A comprising information about file selection 310 and information provided by bundle server 324.

[0038] In block 210, token 314A is delivered to a recipient as token 314B, by a token delivery system 340. Token delivery system 340

delivers token **314A** from computer system **102A** to computer system **102B**. Any suitable mechanism may be used to deliver token **314B** to a recipient. Example delivery methods include attaching token **314A** to an email message or supplying token **314A** by copying it to a portable data store for physical delivery to computer system **102B**.

[0039] In block **212**, the recipient uses computer system **102B** to request redemption of token **314B** by providing token **314B** to a redeemer **330**. Redeemer **330** may present a display of tokens available in data store **110B** on the graphical display monitor **114B**. The recipient identifies token **314B** using the user interface.

[0040] In block **214**, redeemer **330** establishes communication with bundle server **324** to request retrieval of bundle **322**. Redeemer **330** uses information from token **314B** (which was provided by bundle server **324** and stored in token **314A** during block **208**) to identify the required bundle. This information comprises:

- a bundle server communication address, corresponding to the bundle server **324** that stored bundle **322**, and,
- bundle identification information, corresponding to bundle **322**.

In a simple scenario, redeemer **330** establishes communication with bundle server **324** at the bundle server communication address.

[0041] Redeemer **330** requests bundle server **324** to deliver the content of bundle **322** identified by the bundle identification information. Bundle server **324** tests the bundle identification information to determine if it corresponds to bundle **322** that it serves. If the test passes, bundle server **324** replies with the content of bundle **322** in block **216**. Otherwise bundle server **324** refuses the request.

[0042] Access to bundle store 320 and bundle 322 may be limited so that only bundle server 324 has access. Retrieving bundle 322 from bundle server 324 by guessing at the bundle identification information is not practically possible (i.e. is prohibitive), as described below.

5

[0043] In block 216, redeemer 330 retrieves the content of bundle 322 by communicating with bundle server 324. When communication is complete, the retrieved content of bundle 322 is presented for use by the recipient. In some embodiments, redeemer 330 creates folders and files, corresponding to file selection 310, as retrieved files 332 at a predefined location, such as a predefined folder, in data store 110B.

[0044] In block 218, method 200 ends with the sharer having successfully shared file selection 310 with the recipient. One advantage of method 200 is that the effort, required for both sharer and recipient is minimized. Possession of token 314 and access to a redeemer 330 is all that is required for the sharer to obtain bundle 322. Yet, access to bundle 322 is prohibitive without token 314. Another advantage is that the consumption of communication resources required to effect the sharing of data is deferred to a time of each recipient's choosing. Another advantage is that system 300A comprises loosely coupled components, allowing sharer and recipient systems to be dissimilar and allowing a wide range of communication methods, for delivering a token or for delivering the content of bundles to be used.

25

Other Aspects

[0045] Features of methods 200 and systems 300 according to extended embodiments of the invention are described below. These aspects of the system are included to illustrate particular applications of the invention. Systems according to the invention may not have all of the features described below and may be constructed differently from

30

the specific embodiments described below. Figures 4 and 5, described in further detail below, are data structure diagrams identifying the composition of a bundle and a token, respectively, in an example embodiment of system **300**. Data components shown in Figures 4 and 5 are identified by the reference numbers used in Figures 4 and 5 in the following description.

System Configuration Aspects

[0046] System **300** permits a number of possible configurations. System **300A**, which is described above, illustrates one configuration which includes two interacting computer systems. In another configuration, system **300** operates within a single computer system **102A**. Redeemer **330** runs on computer system **102A** to produce retrieved files **332** on data store **110A**. This configuration enables different people using the same computer system **102A** to share files with each other using method **200**.

[0047] In another configuration, illustrated in Figure 3B, the components of a system are distributed among three or more interacting computer systems. This configuration enables people to share files with each other using method **200**, where bundles are stored on a computer system **102C** that is different from the computer systems used by the sharer and recipient. In this configuration, method **200** differs from the basic description as follows:

- In block **206**, tokenizer **316** communicates with bundle server **324**, located on computer system **102C**.
- Bundle server **324** creates bundle store **320** and bundle **322** in data store **110C**.
- In block **214**, redeemer **330** establishes communication with bundle server **324** on computer system **102C**.

[0048] Some embodiments of the invention have multiple bundle servers **324**. In such embodiments, each tokenizer **316** may be configured to interact with a predefined bundle server **324**. In the alternative, a tokenizer **316** may choose from among a plurality of available bundle servers **324**.

File Selection Aspects

[0049] In one embodiment, an application **312A** (e.g. a file system browsing application) identifies file selection **310** and provides it to tokenizer **316**. In general, information described herein as being exchanged through a user interface associated with tokenizer **316** may be exchanged through an interface specific to tokenizer **316** or an interface provided by some application other than tokenizer **316**.

[0050] In a preferred embodiment of block **204**, file selection **310** is examined by tokenizer **316** and file selection **310** is augmented with a detailed list of subfolders and files to form the basis of bundle **322**. Identifying a folder in file selection **310** causes each file, located in the determined folder, to be selected. Similarly, each sub-folder in a hierarchy of sub-folders is examined by tokenizer **316** for files and folders. Tokenizer **316** provides the augmented file selection **310** to bundle server **324**. Bundle server **324** stores information about the augmented file selection **310** as resource items **450** in bundle **322**. Each resource item corresponds to a file or a folder identified by file selection **310**. Resource items **450** provide organization to bundle **322** to facilitate storing information about each file and folder. The information in a token **314** about each file and folder in selection **310** can be useful to the holder of a token, as described below.

[0051] In one embodiment a bundle server can store information about a resource in a bundle using either "copy semantics" or "reference semantics". Tokenizer 316 may provide a user interface which permits the sharer to define which semantics to use. Available
5 semantics are displayed on graphical display monitor 114A. The sharer identifies the desired semantics using the user interface. Tokenizer 316 identifies the semantics to be used to bundle server 324.

[0052] When using copy semantics, bundle server 324 copies
10 certain information about the resource to the bundle. When using reference semantics bundle server 324 omits storing certain information about a resource. Instead, the certain information is obtained when bundle 322 is retrieved, as described below. In some embodiments, tokenizer 316 does not examine and augment the folders in file selection
15 310 when reference semantics are being used. Instead, this examination is deferred until bundle 322 is retrieved.

[0053] For a folder in file selection 310, the information about a resource may comprise; for example:
20 I) a type 452, identifying the folder as a "folder using reference semantics" or "folder using copy semantics";
ii) a pathname 454, identifying the location of the folder in data store 110; and
iii) an attributes data structure 456, that is omitted when using
25 reference semantics, and otherwise comprises:
a) a last modified date 458, identifying the date and time that the contents of the folder was last changed.

[0054] For a file in file selection 310, the information about a
30 resource may comprise, for example, the same information as for a folder but with the following differences:

- I) a type **452**, identifying the folder as a "file using reference semantics" or "file using copy semantics";
- ii) a pathname **454**, identifying the location of the file in data store **110**;
- 5 iii) an attributes data structure **456**, that is omitted when using reference semantics, and otherwise comprises:
- a) a size **457**, identifying the storage allocation required for the file; and
- b) a last modified date **458**, identifying the date and
10 time that the contents of the file was last changed;
and
- iv) when copy semantics are being used, a content **459** which contains the data stored in the file (content **459** is omitted when reference semantics are being used).

15

[0055] In some embodiments only copy semantics are used. In some embodiments only reference semantics are used.

Bundle Creation Aspects

20 **[0056]** Embodiments of the invention may provide controls, which may, for example, be accessed by way of a user interface of tokenizer **316** to enable a sharer to perform one or more of the following functions:

- Identify a particular bundle server **324** to be used for creating
25 bundle **322**;
- Define an expiry period for bundle **322**. The expiry period may support a range of times including identifying that bundle **322** should never expire. It is most practical to allow a bundle never to expire if bundle server **324** is located on the sharer's computer
30 system **102A**.

- Identify a retrieval limit that specifies a maximum number of times that bundle **322** may be redeemed.

Information specified by these controls may be provided to bundle server **324** in block **206**.

5

[0057] In a preferred embodiment, a bundle server **324** provides a plurality of bundle stores. Each bundle store may be manifested as a folder in a data store. Each bundle store may contain one file corresponding to each stored bundle. A bundle server **324** may provide a different bundle store for each of a plurality of sharers authorized to use the bundle server.

[0058] Bundle server **324** is configured to find bundle stores in one or more storage locations. For each identified bundle store, bundle server **324** maintains a bundle store table containing information about the bundle store. The bundle store table may contain entries which are each indexed by one or more organizational attribute values. The one or more organizational attribute values for each entry in the bundle store table are unique. For example, organizational attributes could comprise: sharer identity, sharer employer and sharer department. In some embodiments, sharer identity is the only organizational attribute used.

[0059] When a sharer interacts with tokenizer **316** to identify a file selection **310**, information about the sharer is also obtained. This information may include one or more organizational attributes of the sharer. Tokenizer **316** automatically obtains sharer organizational attributes from processing unit **112A** and may allow the sharer to change the attributes through the user interface. Tokenizer **316** provides the information about the sharer to bundle server **324** which uses the information to search its bundle store table. If a bundle store table entry is found that matches the sharer's one or more organizational attribute

values, the matching bundle store table entry is used. If a matching bundle store table entry is not found, a new table entry is created comprising values corresponding to the sharer's one or more organizational attribute values.

5

[0060] When a matching table entry is not found, bundle server 324 also creates a new folder for bundle store 320 and stores the folder pathname in the new bundle store table entry. When bundle server 324 runs on the sharer's computer system 102, tokenizer 316 may provide a user interface for the sharer to define the bundle store folder pathname. Otherwise the bundle store folder pathname may be generated by bundle server 324 based on the sharer's one or more organizational attribute values.

10 [0061] When a matching table entry is not found, bundle server 324 also generates a bundle store identifier and stores the bundle store identifier in the new bundle store table entry.

[0062] When a bundle store table entry exists, bundle server 324 creates new bundle 322 corresponding to file selection 310. Bundle server 324 generates a unique bundle name, based on file selection 310, and assigns the name to the bundle file as name 460. Bundle server 324 also generates a bundle identifier and stores it in bundle 322 as identifier 410. Bundle server 324 also stores the current date in bundle 322 as creation date 430. Bundle server 324 also generates an expiry date, based on creation date 430 and the bundle expiry period, provided by tokenizer 316. The expiry date is stored in bundle 322 as expiry date 440. Expiry date 440 may be used by bundle server 324 to delete bundle 322 automatically when the current date becomes later than the expiry date. Bundle server 324 also stores the retrieval limit, provided by tokenizer 316 in bundle 322 as retrieval limit 420. Bundle server 324

20
25
30

supplies creation date 430 and expiry date 440 to tokenizer 316 for storing in token 314 as creation date 530 and expiry date 540, respectively.

- 5 [0063] When bundle 322 is created, bundle server 324 also generates additional information associated with aspects of retrieving bundle 322. Bundle server 324 supplies this and previously described information to tokenizer 316 for storing in token 314. This information and its use are detailed below.

10

Token Creation Aspects

- [0064] In a preferred embodiment of block 208, token 314A is a file stored in data store 110A. In response to a bundle creation request, bundle server 324 provides tokenizer 316 with resource items 450, omitting content 459. Tokenizer 316 stores resource items 450 in token 314 as resource items 550 for use during bundle retrieval as described below. Bundle server 324 also provides tokenizer 316 with the sharer organizational attributes which are stored in token 314 as sharer attributes 520.

20

- [0065] In one embodiment, when token 314A is created, the tokenizer user interface enables the sharer to specify a location in data store 110A to store token 314A.

- 25 [0066] In one embodiment, when token 314A is created, tokenizer 316 may optionally, at the sharer's discretion, interact with an application 312A to provide token 314A, to application 312A. As an example, application 312A may be an email client and the interaction requests that the email client create a new email message with token 314A as an attachment.
- 30

Token Redemption Aspects

5 [0067] In one embodiment of block **212**, redeemer **330** automatically attempts to retrieve the entire bundle **322** identified by a token. In another embodiment, redeemer **330** provides a user interface that presents a display of resource items **550** on graphical display monitor **114B**. The display may be generated on the basis of information in the token. The recipient selects one or more resource items, using mouse **116B** or keyboard **118B**, adding each corresponding pathname **553** to a retrieval list.

10

[0068] In one embodiment, application **312B** obtains token **314B** and interacts with redeemer **330** to identify a retrieval list. In general, information exchanged through a redeemer user interface may be exchanged through an application interface.

15

[0069] The retrieval list is provided to bundle server **324** as part of the retrieval request in block **214**. In block **216** bundle server **324** communicates content **459** corresponding to each resource item in the retrieval list.

20

[0070] In one embodiment, redeemer **330** can use expiry date **540** to determine whether bundle **322** can be redeemed without contacting bundle server **324**. Redeemer **330** does not request retrieval of bundle **322** if expiry date **540** has been reached.

25

[0071] When bundle **322** is created using reference semantics, blocks **212** and **214** are modified. First, redeemer **330**, prior to displaying resource items **550**, establishes communication with bundle server **324**. Next, redeemer **330** communicates with bundle server **324** to obtain a current list of resource items **450** from bundle server **324**, (rather than using the resource items **550** from token **314**). Bundle

30

server 324 examines data store 110A corresponding to resource items 450 to augment resource items 450 with current information about folders, subfolders and files. Bundle server 324 communicates the augmented resource items to redeemer 330. Next, redeemer 330
5 presents the augmented resource items in its user interface and the recipient selects one or more resource items. Redeemer 330 captures the selection as a retrieval list. Next, redeemer 330 requests bundle server 324 to retrieve bundle 322, providing the retrieval list. In block 216, bundle server 324 communicates file resource item content 459
10 from data store 110A, using pathname 454, rather than from bundle 322.

[0072] In preferred embodiments of the invention, bundle server 324 automatically processes requests for token redemptions based solely
15 upon information from tokens 314 and does not require separate authentication information from a recipient attempting to redeem a token.

Bundle Server Communication Aspects

20 [0073] System 300 may have a number of optional features relating to the communication of bundles 322. One aspect relates to redeemer 330 locating bundle server 324. In a preferred embodiment, when token 314 is created, bundle server 324 provides the current bundle server computer name and bundle server communication address
25 to tokenizer 316 to be stored in token 314 as bundle server computer name 512 and as one address in bundle server communication addresses 513. However, it is advantageous to be able to reassign computer addresses, change computer names and reallocate bundle stores 320 to different bundle servers 324.

[0074] In some embodiments bundle server **324** maintains a list of historical communication addresses it has used for bundle creations. Bundle server **324** provides these addresses to tokenizer **316** to store in bundle server communication addresses **513**, as possible alternative
5 communication addresses.

[0075] In block **214**, redeemer **330** first attempts to use the bundle server communication addresses **513** to establish communication with bundle server **324**. If communication cannot be established using this
10 method, or if communication is established but the bundle server(s) at communication address(es) **513** no longer provide access to the required bundle store **320**, redeemer **330** uses searching methods to establish communication with a bundle server **324** that does provide access to bundle store **320**. Searching methods can comprise:

- 15 • attempting communication at addresses neighboring addresses included in bundle server communication addresses **513**;
- attempting communication using other addresses previously used by redeemer **330**; and,
- attempting communication using a multicast method.

20 For an attempted communication, redeemer **330** sends a message, including bundle identification, to a candidate bundle server. The candidate bundle server uses at least part of the bundle identification to determine whether it provides access to bundle store **320**. If access to bundle store **320** is available, bundle server **324** replies, indicating a
25 successful attempt, and redeemer **330** establishes communication. If searching methods fail a configured relay service may be used to establish communication as described below.

[0076] When communication with bundle server **324** is
30 established, redeemer **330** requests retrieval of bundle **322**, providing the bundle identification. Bundle server **324** uses at least part of the

bundle identification to locate bundle **322**. If a bundle **322** is located, bundle server **324** transfers the contents of bundle **322** to redeemer **330**. If bundle server **324** cannot locate a bundle **322** which matches the bundle identification than bundle server **324** refuses the request in a
5 reply to redeemer **330**.

[0077] Another aspect relates to communication disruptions that may occur while redeemer **330** is retrieving bundle **322** from bundle server **324**. Redeemer **330** is able to resume a communication, during
10 block **216**, by creating a retrieval session and retrieving bundle **322** in parts. If a disruption occurs, redeemer **330** identifies the last fully received part of bundle **322** and provides that information to bundle server **324** along with a request to resume retrieval of bundle **322**.

15 [0078] Some embodiments of the invention provide a relay service to facilitate communication between redeemer **330** and bundle server **324** when bundle **322** is retrieved. Such a relay service can permit operation when one or both of redeemer **330** and bundle server **324** are not permitted to accept unsolicited communications from outside their
20 computer system **102** or sub-network **104**. This is common when firewalls are used in computer systems **102** or sub-networks **104**.

[0079] Figure 6 is a block diagram illustrating a relay service **350** corresponding to one embodiment of system **300**. System **300** may
25 include zero or more relay services **350**. A relay service **350** comprises a number of relay elements, including zero or one connection distributor **652**, one or more connection brokers **654**, and one or more transfer agents **656**. Each of these relay elements can run on one of a plurality of computer systems **102**. Configuration options range from a single
30 computer system **102**, hosting all of the relay elements, to a separate

computer system **102** for each element. The function of each relay element is described below.

[0080] Connection Distributor **652**, runs on a computer system **102**
5 connected via a sub-network **104** having security provisions that allow it to receive unsolicited communications. When bundle server **324** starts, it establishes communication with a relay service **350**, if it has been configured to do so. Bundle server **324** automatically initiates communication with an address which corresponds to connection
10 distributor **652**, if one exists in relay service **350**, or to connection broker **654** otherwise. Connection distributor **652** maintains a list of connection brokers **654** and assigns each bundle server **324** to a connection broker **654** to distribute communication load among connection brokers **654**. Bundle Servers **324** communicate with their
15 assigned connection broker **654** directly, after receiving an assignment from connection distributor **652**. For example, in Figure 6, bundle server **324A** is assigned to connection broker **654A**. Similarly bundle servers **324B** and **324C** are assigned to connection broker **654B**.

20 **[0081]** Connection Broker **654**, runs on a computer system **102** connected via a sub-network **104** having security provisions that allow it to receive unsolicited communications. When bundle server **324** establishes communication with a connection broker **654**, bundle server **324** provides connection broker **654** with information comprising a list
25 of identifiers corresponding to bundle stores that bundle server **324** serves. Connection broker **654** maintains the information supplied by bundle server **324** for use when a redemption request is received. Periodically and when a bundle store is added or removed, bundle server **324** communicates with connection broker **654** to provide
30 updated information.

[0082] In block 214 of method 200, when redeemer 330 establishes communication with bundle server 324A, redeemer 330 first attempts to directly communicate with bundle server 324. If that attempt fails, redeemer 330 then attempts to establish communication with a relay service 350 using relay service communication address 514, providing information which comprises the bundle store identifier 511. If connection distributor 652 exists, communication address 514 corresponds to connection distributor 652. Connection distributor 652 replies to redeemer 330 with an indication that redeemer 330 should communicate with connection broker 654A, previously assigned to broker requests for bundle server 324A. Otherwise, the communication address 514 corresponds to a single connection broker 654A.

[0083] When connection broker 654A receives the redemption request from redeemer 330, connection broker 654A determines if identifier 511 corresponds to one served by a communicating bundle server 324A. If a bundle store identifier is matched, connection broker 654A allocates a new transfer session and assigns it to a transfer agent 656A. Connection broker 654A replies to redeemer 330, providing information about the transfer session, transfer agent 656A, and a time period to wait before attempting communication with transfer agent 656A. The time period can be based on the time when bundle server 324A is expected to next communicate with connection broker 654A. When bundle server 324A next communicates with connection broker 654A, connection broker 654A replies with information about the pending redemption request, the allocated transfer session and the assigned transfer agent 656A. The communication between redeemer 330 and bundle server 324A is then handled by transfer agent 656A.

[0084] A transfer agent 656, runs on a computer system 102 connected via a sub-network 104 having security provisions that allow

transfer agent **656** to receive unsolicited communications from at least bundle server **324** and redeemer **330**. Transfer agent **656** effects bi-directional communication by buffering a request and forwarding it to the receiver when the receiver polls transfer agent **656**. Both bundle
5 server **324** and redeemer **330** are configured to periodically poll transfer agent **656A** during a transfer session.

[0085] A connection broker **654** maintains a list of available transfer agents **656** and dynamically assigns redemption requests to
10 transfer agents **656** to distribute workload among transfer agents **656**. For example, in Figure 6, redeemer **330** is using transfer agent **656B** in a redemption involving bundle server **324B** and transfer agent **656C** in a redemption involving bundle server **324C**.

15 **Bundle Retrieval Aspects**

[0086] In a preferred embodiment, bundle server **324**, having received a redemption request for bundle **322**, first obtains retrieval count **470** and retrieval limit **420** for the bundle. If retrieval count **470** is less than retrieval limit **420**, the request is processed and retrieval count
20 **470** is incremented. Otherwise, the retrieval request is refused.

[0087] In some embodiments, the redeemer user interface enables the recipient to define the location in data store **110B** for storing retrieved files **332**. The default location for locating the retrieved files
25 can be configured to depend on information supplied to redeemer **330**. For example, when token **314B** is supplied to redeemer **330** from data store **110B**, the default location for the retrieved files could be the same folder where token **314B** is located. When token **314B** is supplied to redeemer **330** from an application (e.g. an email application) the system
30 may be configured to prompt the recipient for a location or to use a previously specified location.

[0088] In another embodiment, redeemer 330 can be configured to deliver the files to an application 312B.

- 5 [0089] In another embodiment, redeemer 330 user interface can inform a recipient that there has been a change in file selection 310 corresponding to bundle 322 that is represented by the recipient's token 314. In block 206, when copy semantics are used, bundle server 324 computes and provides tokenizer 316 with a digest of bundle 322.
- 10 Tokenizer 316 stores the digest 568 in token 314. In block 216, redeemer 330 re-computes a digest using the retrieved bundle 322 to determine whether bundle 322 has changed since token 314 was generated. Redeemer 330 displays an indication of the digest comparison to the recipient. In another embodiment, a digest can be
- 15 associated with each resource instead of the bundle to allow digests to be more useful for selective retrieval.

Security Aspects

- [0090] System 300 may have a number of features relating to the security of bundles and tokens and the privacy of sharers and recipients.
- 20

[0091] In one embodiment the bundle identification information, generated by bundle server 324 for new bundle 322, comprises:

- the bundle store identifier, stored in token 314 as identifier 511;
- 25 • the bundle identifier, stored in token 314 as identifier 562; and,
- an encrypted bundle name, stored in token 314 as encrypted name 566.

The bundle store identifier and the bundle identifier have values whose range of possible values is substantially large and whose values have

30 been generated by a cryptographically strong random or pseudo-random

method. The encrypted bundle name is an encrypted form of the bundle name, encrypted with the bundle store private key.

[0092] The bundle identification information is substantially
5 unguessable. “Unguessable” means that successfully querying bundle
server 324, to obtain a bundle 322 without having the bundle
identification information for bundle 322, would require computer
processing power and elapsed time large enough to make guessing
prohibitive. The range of possible values for bundle identifiers and
10 bundle store identifiers can be sized to provide the level of security and
privacy desired. The unguessability of the bundle identifier is of greater
importance in situations where a violator may attack a bundle store
whose bundle store identifier is known by decoding a token associated
with a different bundle in the same bundle store.

15
[0093] In practice, the range of possible values for the bundle
identifier and/or the bundle store identifier can be chosen so that the
probability of correctly guessing a valid bundle identifier/bundle store
identifier is low enough to provide an acceptable level of security (e.g.
20 one in a million guesses). The probability of correctly guessing bundle
identification information corresponding to a bundle is a function of the
number of bundles accessed by a bundle server and the range of
possible values for the bundle identification information. The security of
any bundle is a function of the probability of a correct guess and the
25 number of guesses that can be made.

[0094] the request rate can be governed by the bundle server.
Rates on the order of 10^6 requests per second can be used as an example
limit. One may assume that a violator will not be willing to continue
30 guessing for more than one year. One may also assume that one bundle
server provides access to at most 10^6 bundles. Assume a value range on

the order of 10^{30} values. This gives a probability of 10^{-24} that a single guess of some bundle accessed by a bundle server is correct. In one year, a violator could make approximately 10^{13} guesses. Depending on the context of its use, this may be an acceptable level of security. The request rate and value ranges for the bundle identification information may be adjusted to determine a suitable level of unguessability.

[0095] The bundle and/or bundle store identifier value range may be greater than or equal to 10^{10} , greater than or equal to 10^{20} , greater than or equal to 10^{30} or greater than or equal to 10^{40} . The security of a system according to the invention against attempts to guess bundle identification information can be increased by maintaining a large ratio of possible values for bundle identification information to a number of bundles in a bundle store. In some embodiments of the invention this ratio equals or exceeds $10^{15}:1$. For more security the ratio may equal or exceed, for example, $10^{20}:1$, $10^{24}:1$ or $10^{30}:1$. The ratio may exceed a maximum number of requests for bundles that could be made in one year at a maximum request rate of the bundle server by a factor of at least 1000, or, for greater security, for example, a factor of 10^6 , 10^{10} , or more.

[0096] In some embodiments, random values for bundle identifiers and/or bundle store identifiers are generated using the pseudo-random method of SunTMJavaTMs SecureRandom class, using the "SHA1PRNG" algorithm from the "SUN" cryptographic service provider. In other embodiments, pseudo-random generators having the following properties that may be equivalent to or better than SunTMJavaTMs SecureRandom class can be used to generate random identifiers:

- multiple input sources;
- input source mixing strength;
- input data analysis resistance;

- input data manipulation resistance;
- output data analysis resistance; and
- internal state analysis resistance.

5 **[0097]** In some embodiments, public/private key pairs are associated with bundle stores **320** and/or with individual bundles **322**. In some embodiments these public/private key pairs are generated at the time a bundle or bundle store is created. For example, upon creating a new bundle store, bundle server **324** may generate a bundle store key pair and store the bundle store key pair in a new bundle store table entry
10 corresponding to the new bundle store. The bundle store key pair is part of an asymmetric cryptographic system, whereby data encrypted with the bundle store private key may be decrypted using the bundle store public key and data encrypted with the bundle store public key may be
15 decrypted by the bundle store private key but not with the bundle store public key.

[0098] In one embodiment bundle store private and public keys are generated using methods conforming to standard RSA PKCS #1
20 (RFC3447). In other embodiments, bundle store private and public keys are generated using methods providing equivalent or better cryptographic strength than RSA PKCS #1.

[0099] In one embodiment bundle server **324** generates a unique
25 bundle key for bundle **322** during block **206**. The bundle key may be generated as part of a symmetric cryptographic system, complying with US Federal Information Processing Standard FIPS-197, whereby data encrypted with the bundle key can be decrypted with the bundle key. The bundle key can be used to encrypt content **459** corresponding to
30 each resource item stored in bundle **322**, or corresponding to each resource item retrieved from file store **110A** when using reference

semantics. The bundle key is provided to tokenizer **316** for storing in token **314** as key **564**. Encrypting content **459** protects the privacy of the sharer's information while stored in bundle store **320** and during retrieval.

5

[0100] In one embodiment a sharer can provide additional security for token **314** and bundle **322** by providing a pass-phrase to tokenizer **316**. Tokenizer **316** enables a sharer to supply a pass-phrase during block **204**. Tokenizer **316** encrypts token **314** using the pass-phrase during block **208**. During block **212** the recipient must supply the pass-phrase in order for redeemer **330** to successfully decrypt the content of token **314**.

[0101] In one embodiment a recipient's privacy can be increased during retrieval of bundle **322**. The recipient's attributes (e.g. computer account name), may be automatically obtained by redeemer **330** from processing unit **112B**, and may be changeable by the recipient through the redeemer user interface. The recipient's attributes may be provided to bundle server **324** by redeemer **330** in block **216**. Redeemer **330** encrypts the recipient's attributes using public key **515**, corresponding to bundle store **320**, stored by tokenizer **316** in block **208**. In one embodiment this method can be used to ensure that all communication from redeemer **330** to bundle server **324** is private.

25 **Management Aspects**

[0102] System **300** may have a number of aspects relating to the management of system **300** and the bundles it administers. One aspect relates to producing system **300** usage information. In one embodiment, bundle server **324** maintains a record of redemption requests for bundle **322**. The record, for each request, can include information about the recipient and/or the recipients' computer system **102B**, provided by

30

tokenizer **316**, and information about the bundle retrieval process, provided by bundle server **324** and/or relay service **350**. In one embodiment this information is stored in data store **110** so that sharers may see who has redeemed their tokens. In another embodiment, the
5 system can provide an application for a person to obtain redemption information, pertaining to bundle **322** corresponding to their file selections **310**, on request or automatically.

[0103] System **300** may restrict usage or provide information to
10 support billing for usage. In one embodiment, a relay service records the volume of data communicated by its transfer agents. The relay service can accumulate this information in groupings based on a bundle store, a bundle server or other information. A relay service can have thresholds configured for each grouping of data volume metrics, which
15 may be used to refuse a redeemer's request or to trigger generation of information to be used for billing. In another embodiment, a bundle server can filter retrieval requests based on quotas applied to the volume of information communicated. Quotas can be established on variety of bases, examples including: a bundle and a bundle store.

20

Other Embodiments

[0104] The following descriptions illustrate other embodiments of the system **300** and method **200**. These are provided to illustrate variations of the invention but do not limit the scope of the invention.

25

[0105] In one embodiment the invention is integrated with an email delivery system as illustrated in Figure 3. For this aspect of system **300**, application **312A** scans email messages, queued for delivery, for file attachments. Each email message may include zero, one or multiple
30 attached files. Attached files, whose sizes, or whose aggregate sizes, exceed a configured threshold, are automatically placed in file selection **310** by the email application **312A** and information about file selection

310 is provided to tokenizer **316**. Tokenizer **316** generates a token **314** corresponding to the attachments. The attachments are automatically stored in a bundle **322** in a bundle store **324**. The email application **312A** substitutes the file attachments in the email message with token **314**, provided by tokenizer **316**, prior to delivering the message.

[0106] In this embodiment, token delivery system **340** is manifested by the transmission and reception parts of the email system. On receipt of an email message, token **314B** and its association with the corresponding email message are stored in data store **110B** by the reception part of the email system. The recipient's email application **312B** provides a method for displaying the email message and associated token **314B**, which may then be selected. Token **314B** may be of a file type that is associated with redeemer **330** so that, when selected, token **314B** is automatically provided to redeemer **330** for retrieval of the original message attachments.

[0107] In other embodiments, a sharer may use a bundle manager to view and alter the content of bundle **322**. As an example, the bundle manager can enable a sharer to substitute an original version of a resource item in bundle **322** with a newer version. Token **314B** can still be used to obtain bundle **322**. A digest comparison can indicate that the retrieved content is different than the original.

[0108] In other embodiments, the organizing structure, content, and format of a bundle store, a bundle and a token can change to enhance or optimize certain aspects of the invention. Examples include: storing a bundle in a database instead of a file, compressing data to save storage space, eliminating elements to save storage space, adding elements to correspond with additional aspects, and changing elements to correspond with different data organizing methods or different communication methods. Examples include adding additional contextual

information to a token comprising: low-resolution preview data corresponding to high-resolution image data from a file selection, annotations pertaining to a file selection, and references to other data potentially relevant to a file selection. In general, contextual information
5 can be associated with the file selection, a resource item or by an arbitrary topic referencing one or more resource items. In one embodiment, a tokenizer may analyze the file selection to derive contextual information. In another embodiment, a tokenizer may provide a user or application for providing contextual information. As
10 an example, a sharer may wish to provide annotations about the selection of files or about specific files. Since the contextual information is in the token, a recipient can review the contextual information before deciding whether to redeem the token. A recipient may use the contextual information to select a subset of resources to download.

15
[0109] Certain implementations of the invention comprise computer processors which execute software instructions which cause the processors to perform a method of the invention. For example, tokenizer **316**, bundle server **324**, and redeemer **330** may all be
20 implemented by providing software which runs on one or more computer systems **102** and causes the computer systems to operate according to methods described above. The invention may also be provided in the form of a program product. The program product may comprise any medium which carries a set of computer-readable signals
25 comprising instructions which, when executed by a computer processor, cause the data processor to execute a method of the invention. The program product may be in any of a wide variety of forms. The program product may comprise, for example, physical media such as magnetic data storage media including floppy diskettes, hard disk
30 drives, optical data storage media including CD ROMs, DVDs, electronic data storage media including ROMs, flash RAM, or the like or transmission-type media such as digital or analog communication

links. The instructions may optionally be compressed and/or encrypted on the medium.

5 **[0110]** As will be apparent to those skilled in the art in the light of the foregoing disclosure, many alterations and modifications are possible in the practice of this invention without departing from the spirit or scope thereof. Accordingly, the scope of the invention is to be construed in accordance with the substance defined by the following claims.